Power-gating technique for network-on-chip buffers

M.R. Casu, M.K. Yadav and M. Zamboni

A new approach to reducing leakage power in network-on-chip buffers is presented. The non-uniformity of buffer utilisation is leveraged across the network and power-gating is applied to scarcely utilised buffers. Instead of turning-off the buffers completely, a buffer portion is kept turned-on. This design choice has a significant performance benefit because the buffer is always able to receive network packets. Design aspects and trade-offs in a 45 nm CMOS technology are discussed and results obtained over video application benchmarks are presented. It is shown that it is possible to reduce buffer leakage by 40% without performance penalty.

Introduction: The network-on-chip (NoC) approach uses packet switching to implement an efficient communication infrastructure for systems-on-chip. Switches are a NoC's essential components as they perform packet routing and implement control flow policies at the 'flit' level, flits being the elementary units NoC packets are made of. Switches contain input buffers to store those flits that cannot be immediately delivered at the switch output ports. Since buffers consume the largest part of a NoC's leakage power, up to 64% according to Chen and Peh [1], investigating how to reduce such leakage is a relevant research problem. Fortunately, utilisation of buffers in a NoC varies significantly across a network and buffers that are scarcely utilised can be turned-off via power-gating. Power-gating consists in switching-off an NMOS footer (or a PMOS header) connected in series with the logic in order to cut the leakage current flow.

We propose a novel power-gating technique for NoC buffers using NMOS footers. Instead of turning the entire buffer off, which would negatively affect performance, we keep a portion of the buffer on, so that incoming flits can always be stored. Differently from other approaches, we do not modify the communication protocol between switches and do not add any new signal to the links connecting adjacent switches. We quantify the various overheads of power-gating, which other authors often overlook. We report on design choices and trade-offs in a 45 nm 1.15 V CMOS technology, and discuss power and performance results obtained over a set of benchmarks from the field of video applications.

Comparison with prior art: In [1], Chen and Peh assume that buffers consist of SRAMs, which is correct for large buffers in chip-to-chip networks. On-chip buffers are instead usually synthesised from a standard-cell library, which is our hypothesis. Moreover, only leakage saving and a NoC performance are reported in [1], whereas aspects such as area and energy overhead of power-gating and the impact of power-gating on clock frequency, which we cover in our contribution, are not discussed.

This latter aspect is discussed in [2], but power-gating is used aggressively to completely turn-off a buffer. This choice minimises leakage power, but may negatively affect performance, as flits arriving at a switch's inputs when the buffer is off have to be pushed back and delayed. We avoid this problem by keeping one portion of the buffer always on.

The work in [3] proposes to control each buffer entry via power-gating, but does not quantify the overhead of such approach. Moreover, the conventional credit-based flow control has to be modified, a 'congestion' detection circuit added to the router logic and a congestion signal added to the links connecting switches together.

In [4], Matsutani *et al.* evaluate various power-gating policies, which require special wake-up signals added to the links. Moreover, their power-gating method is based on a modified standard-cell library of CMOS gates, whereas we use a regular library. They evaluate the overhead of power-gating on power, but not the impact on clock frequency, which instead we do.

Proposed design: We started from a 5-port switch based on wormhole and *XY* routing, described in [5] and illustrated in Fig. 1. Four input and output ports (North, East, South and West) connect the switch to four other switches to create a mesh topology. The fifth input and output ports (local) connect it to a processing element. Input ports each contain a circular buffer to store incoming flits. We modified the buffer by partitioning it in banks, each having an NMOS footer for the implementation of power-gating. In the example in Fig. 1, each buffer is made of four banks.

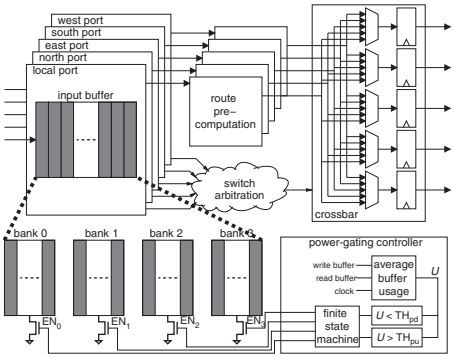

Fig. 1 *Microarchitecture of the switch, with detail of the power-gating technique applied to the input buffers partitioned in banks*

The power-gating signals ($EN_{0 \div 3}$) are driven by a power controller that keeps track of the average buffer usage. In the case of high utilisation, all the banks are powered-on. In the case of low utilisation, only the bank that is currently used is kept on, whereas the other banks are power-gated. The finite-state machine in Fig. 1 makes sure that power-gating is applied only after all flits stored in those banks that are going to be turned-off have been removed from the buffer and delivered at the switch output ports.

There is a trade-off between leakage-power saving and performance. Keeping one bank turned-on is beneficial for network performance, because the input buffer keeps storing flits when the other banks are turned-off. Moreover, differently from other more aggressive approaches (e.g. [3, 4]), there are no modifications to the way switches work and communicate. On the other hand, leakage saving is less than if the buffer was entirely turned-off. Leakage could be minimised by using a small bank, but this would lead to more frequent on–off switching: a small bank easily gets full if utilisation increases and the other banks need to be turned-on to store incoming flits. We ran various simulation experiments using realistic traffic generated by video applications such as MPEG4, VOPD, PIP and MWD – already used in the context of a NoC [6, 7] – and found that four banks with eight entries each are a good trade-off.

The controller compares the average buffer utilisation (computed with a simple cumulative moving average filter) with two thresholds, one for power-down (TH_{pd}) and one for power-up (TH_{pu}). Therefore the controller belongs to the class of on–off controller with hysteresis (bang–bang controller). If a single threshold is used, a much more frequent on–off switching occurs when buffer utilisation has little variations above and below the threshold. The choice of the thresholds is critical for both power and performance. We obtain the best results in our simulations by setting $TH_{pd} = 1$ and $TH_{pu} = 2.5$.

The size of the NMOS footers is another critical design parameter. A small size corresponds to less gate capacitance and so to faster on–off switching and less energy. However, it also entails a higher V_{DS} voltage drop when the bank is active; the consequence is a lower supply voltage for the bank ($V_{dd} - V_{DS}$) and so a longer propagation delay of logic gates and memory elements inside the bank. To determine the best size of the NMOS transistors, we ran SPICE simulations with the post-synthesis netlist of the switch, to which we added the NMOS footers. In our 1.15 V 45 nm technology, the switch's critical path dictates a clock frequency of 1 GHz (at $T = 25$ °C, nominal process). We set the NMOS size such that the critical path delay increases by no more than 1%. In our technology, this corresponds to a 126 µm MOS channel width. The energy required per switch is 0.67 pJ. We show in the following Section that this overhead is negligible compared to power saving.

The turn-on delay of a bank also depends on the NMOS size and in this case it is about 3 ns, which corresponds to three cycles at 1 GHz. To account for variations (e.g. process, voltage and temperature), the controller waits four cycles before enabling read/write operations in the turned-on banks. To limit the impact of turn-on delay, the active bank must not be full when the other banks are being activated, so that it can absorb flits arriving in the meantime. This problem is solved with

a power-up threshold TH_{pu} sufficiently smaller than the bank size (2.5 against 8 in our design).

The area occupied by the switch is 0.109 mm^2 and the area overhead of the NMOS footers is only 0.1%. The area overhead of the power controller is $< 0.02\%$, as well as the additional leakage and dynamic power.

Experimental results: To understand how much leakage power can be saved, we look at the average utilisation of buffers in a typical scenario. Fig. 2 illustrates a NoC mesh arrangement of 12 switches connected locally to processing elements that execute each a different task of the MPEG4 application. Labels within the switches represent task names [6]. The shading represents traffic load – the darker the higher. Numbers indicate average utilisation of input buffers. Only a few buffers have utilisation $> 1\%$, notably those of the switches with the highest traffic load. Therefore, we can apply power-gating to the majority of the buffers. This observation holds also in the three other video applications that we examined: VOPD (12-switch NoC), MWD (12-switch NoC) and PIP (8-switch NoC), not reported for brevity.

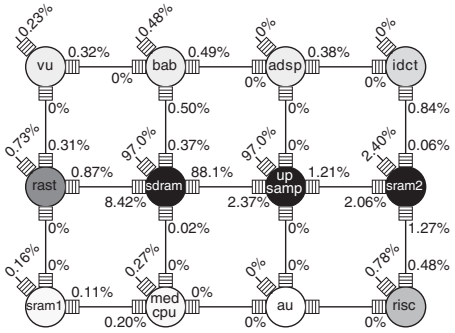

Fig. 2 *NoC mesh for MPEG4 application*

Shading indicates traffic load of switches. Numbers represent average occupation of input buffers. Labels describe task performed in each node [6]

We measured the total NoC leakage with accurate post-synthesis gate-level simulations that include the effect of traffic and buffer utilisation in the four video applications. The histogram in Fig. 3*a* shows that in all applications leakage is about 40% less than in a case without power-gating. (Leakage was estimated at $T = 25$ °C.)

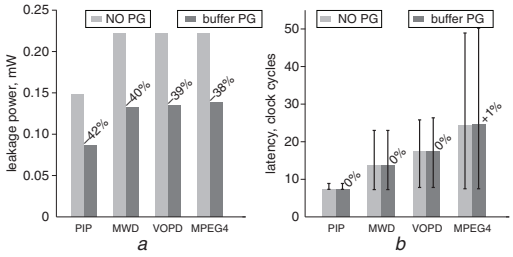

Fig. 3 *Leakage and latency in four video applications: comparison between switch without power-gating and one with power-gating*

a Leakage
b Latency

We also measured the flit latency from source to destination. The histogram in Fig. 3*b* reports average, minimum and maximum latency. Minimum and average latency do not practically increase. Maximum latency increases slightly, but the fact that the average is not affected means that the latency increase is experienced only by a negligible fraction of flits, namely those that find a bank full in the middle of a turn-on event and therefore have to be delayed.

Our experiments show that power switching events do not occur frequently. The distribution of switching events per buffer depends on the application and on the local traffic load. A few switches undergo tens of switching events in 10 000 clock cycles of simulation (up to 61 for a single switch in VOPD), but the average values are only 2.6 events in MPEG4, 2.1 in VOPD, 1.2 in MWD and 1.0 in PIP. With a 1 GHz clock cycle, and with an energy of 0.67 pJ per switching event, the average power overhead (total energy divided by total time multiplied by the number of NoC switches) is 2.1 µW in MPEG4, 1.7 µW in VOPD, 0.96 µW in MWD and 0.54 µW in PIP. If we compare these values with leakage in Fig. 3*a*, we realise that the overhead is negligible (<1% in the worst case).

Conclusion: The leakage power in NoC switch buffers can be minimised with power-gating by exploiting their non-uniform utilisation. Even when some of the NoC nodes are heavily utilised, others are almost idle. Through a partial power-gating approach, it is possible to reduce leakage without incurring a latency penalty. With our approach, we reach leakage saving in buffers of the order of 40% with a negligible impact on a NoC latency, as measured in realistic video applications.

27 September 2013
doi: 10.1049/el.2013.3225

M.R. Casu, M.K. Yadav and M. Zamboni (*Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy*)

E-mail: mario.casu@polito.it

References

1 Chen, X., and Peh, L.-S.: 'Leakage power modeling and optimization in interconnection networks'. Proc. ISLPED'03, Seoul, South Korea, August 2003, pp. 90–95
2 Vangal, S., *et al.*: 'An 80-tile sub-100-W teraFLOPS processor in 65-nm CMOS', *IEEE J. Solid State Chem.*, 2008, **43**, (1), pp. 29–41
3 Kim, G., Kim, J., and Yoo, S.: 'FlexiBuffer: reducing leakage power in on-chip network routers'. Proc. DAC'11, New York, USA, June 2011, pp. 936–941
4 Matsutani, H., *et al.*: 'Performance, area, and power evaluations of ultrafine-grained run-time power-gating routers for CMPs', *IEEE J. Technol. Comput. Aided Des.*, 2011, **30**, (4), pp. 520–533
5 Tota, S., Casu, M.R., and Macchiarulo, L.: 'Implementation analysis of NoC: a MPSoC trace-driven approach'. Proc. GLSVLSI'06, Philadelphia, PA, USA, May 2006, pp. 204–209
6 Bertozzi, D., *et al.*: 'Noc synthesis flow for customized domain specific multiprocessor systems-on-chip', *IEEE TPDS*, 2005, **16**, (2), pp. 113–129
7 Yadav, M.K., Casu, M.R., and Zamboni, M.: 'LAURA-NoC: local automatic rate adjustment in network-on-chips with a simple DVFS', IEEE Trans. Circuits Syst. II, in press